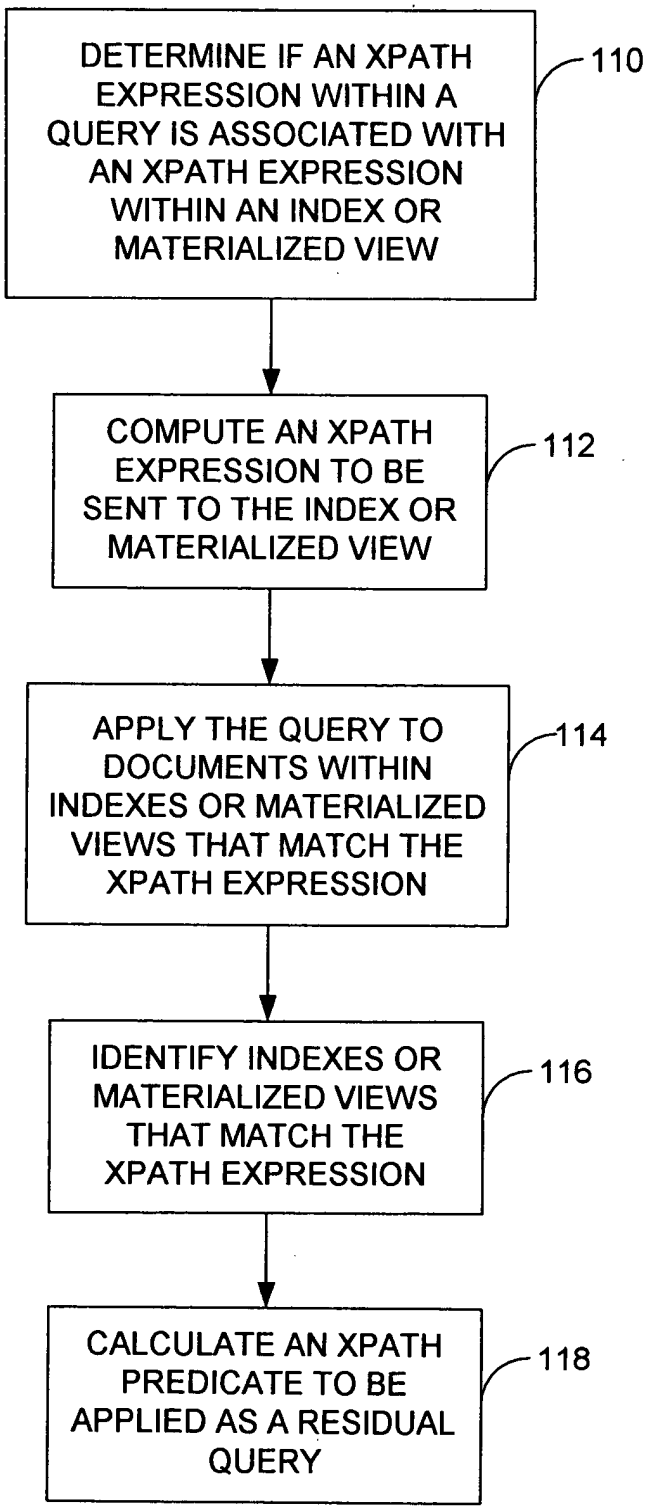

DETERMINE IF AN XPATH EXPRESSION WITHIN A QUERY IS ASSOCIATED WITH AN XPATH EXPRESSION WITHIN AN INDEX OR MATERIALIZED VIEW
110
COMPUTE AN XPATH EXPRESSION TO BE SENT TO THE INDEX OR MATERIALIZED VIEW
112
APPLY THE QUERY TO DOCUMENTS WITHIN INDEXES OR MATERIALIZED VIEWS THAT MATCH THE XPATH EXPRESSION
114
IDENTIFY INDEXES OR MATERIALIZED VIEWS THAT MATCH THE XPATH EXPRESSION
116
CALCULATE AN XPATH PREDICATE TO BE APPLIED AS A RESIDUAL QUERY
118

FIG. 1

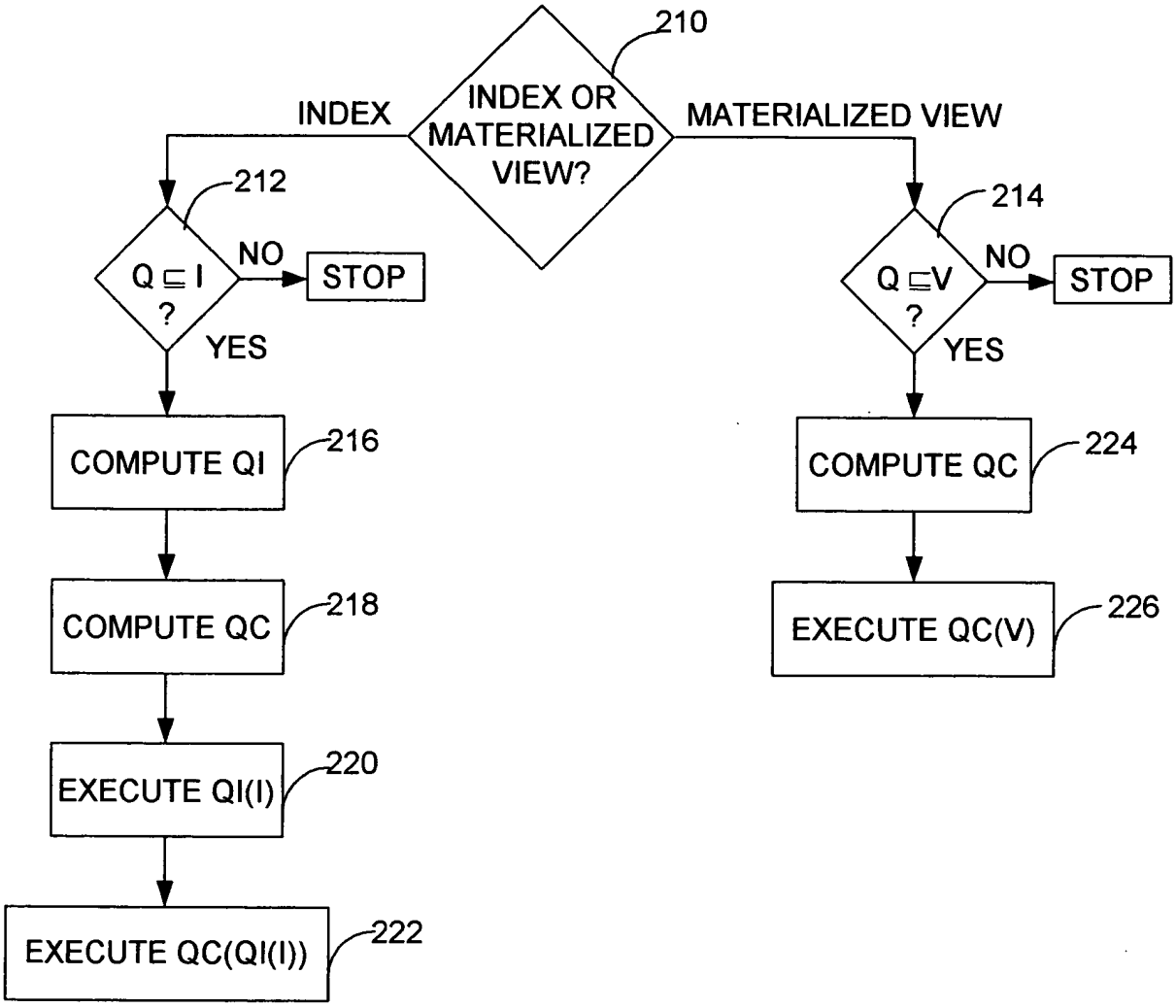

210
INDEX OR MATERIALIZED VIEW?
INDEX
MATERIALIZED VIEW
212
Q ⊑ I ?
NO
STOP
YES
214
Q ⊑ V ?
NO
STOP
YES
COMPUTE QI
216
COMPUTE QC
218
EXECUTE QI(I)
220
EXECUTE QC(QI(I))
222
COMPUTE QC
224
EXECUTE QC(V)
226

FIG. 2

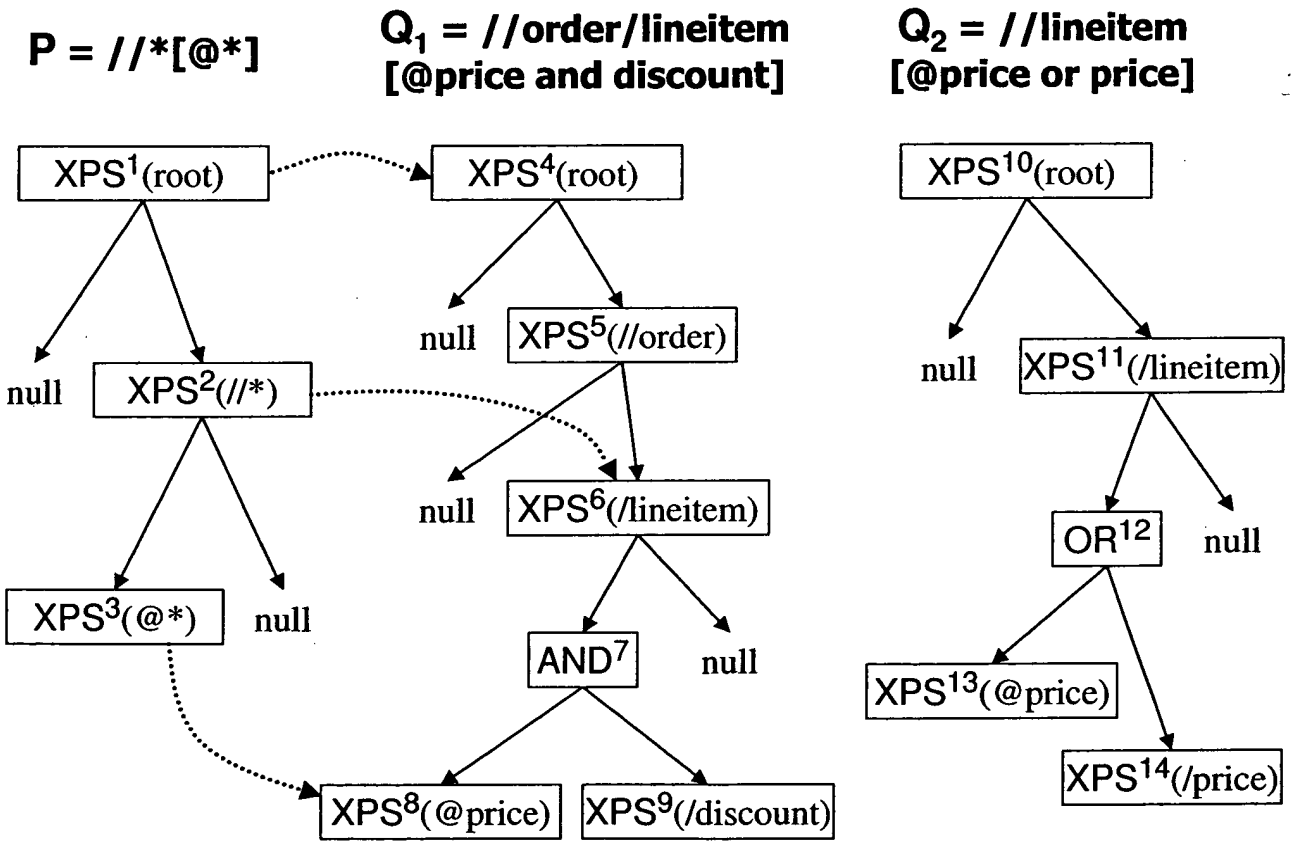

P = //*[@*]
Q1 = //order/lineitem [@price and discount]
Q2 = //lineitem [@price or price]
XPS1(root)
null
XPS2(//*)
XPS3(@*)
null
XPS4(root)
null
XPS5(//order)
null
XPS6(/lineitem)
AND7
null
XPS8(@price)
XPS9(/discount)
XPS10(root)
null
XPS11(/lineitem)
OR12
null
XPS13(@price)
XPS14(/price)

FIG. 3

1	$matchStep(p, q)$	
1.1	if $(q = q_1 \wedge q_2)$	$matchStep(p, q) \rightarrow matchStep(p, q_1) \vee matchStep(p, q_2)$
1.2	else if $(q = q_1 \vee q_2)$	$matchStep(p, q) \rightarrow matchStep(p, q_1) \wedge matchStep(p, q_2)$
1.3	else if $(p_{axis}$ = "descendant")	$matchStep(p, q) \rightarrow \bigvee \{matchChildren(p, c)\}$, $\forall c \in$ { preorder traversal of q}, such that $c_{axis} \neq$ "attribute"
1.4	else if $(p_{axis} = q_{axis})$	$matchStep(p, q) \rightarrow matchChildren(p, q)$
1.5	else	$matchStep(p, q) \rightarrow False$
2	$matchChildren(p, q)$	
2.1	if $(p_{test}$ = "*") $\vee$ $(p_{test}=q_{test})$	$matchChildren(p, q) \rightarrow matchPred(p_{pred}, q) \wedge matchNext(p_{next}, q))$
2.2	else	$matchChildren(p, q) \rightarrow False$
3	$matchPred(p_{pred}, q)$	
3.1	if $(p_{pred}$ = null)	$matchPred(p_{pred}, q) \rightarrow True$
3.2	else if $(q$ = null)	$matchPred(p_{pred}, q) \rightarrow False$
3.3	else if $(p_{pred} = p_1 \wedge p_2)$	$matchPred(p_{pred}, q) \rightarrow matchPred(p_1, q) \wedge matchPred(p_2, q)$
3.4	else if $(p_{pred} = p_1 \vee p_2)$	$matchPred(p_{pred}, q) \rightarrow matchPred(p_1, q) \vee matchPred(p_2, q)$
3.5	else	$matchPred(p_{pred}, q) \rightarrow matchStep(p_{pred}, q_{pred}) \vee matchStep(p_{pred}, q_{next})$
4	$matchNext(p_{next}, q)$	
4.1	if $(p_{next}$ = null)	$matchNext(p_{next}, q) \rightarrow True$
4.2	else if $(q$ = null)	$matchNext(p_{next}, q) \rightarrow False$
4.3	else	$matchNext(p_{next}, q) \rightarrow matchStep(p_{next}, q_{pred}) \vee matchStep(p_{next}, q_{next})$

FIG. 4

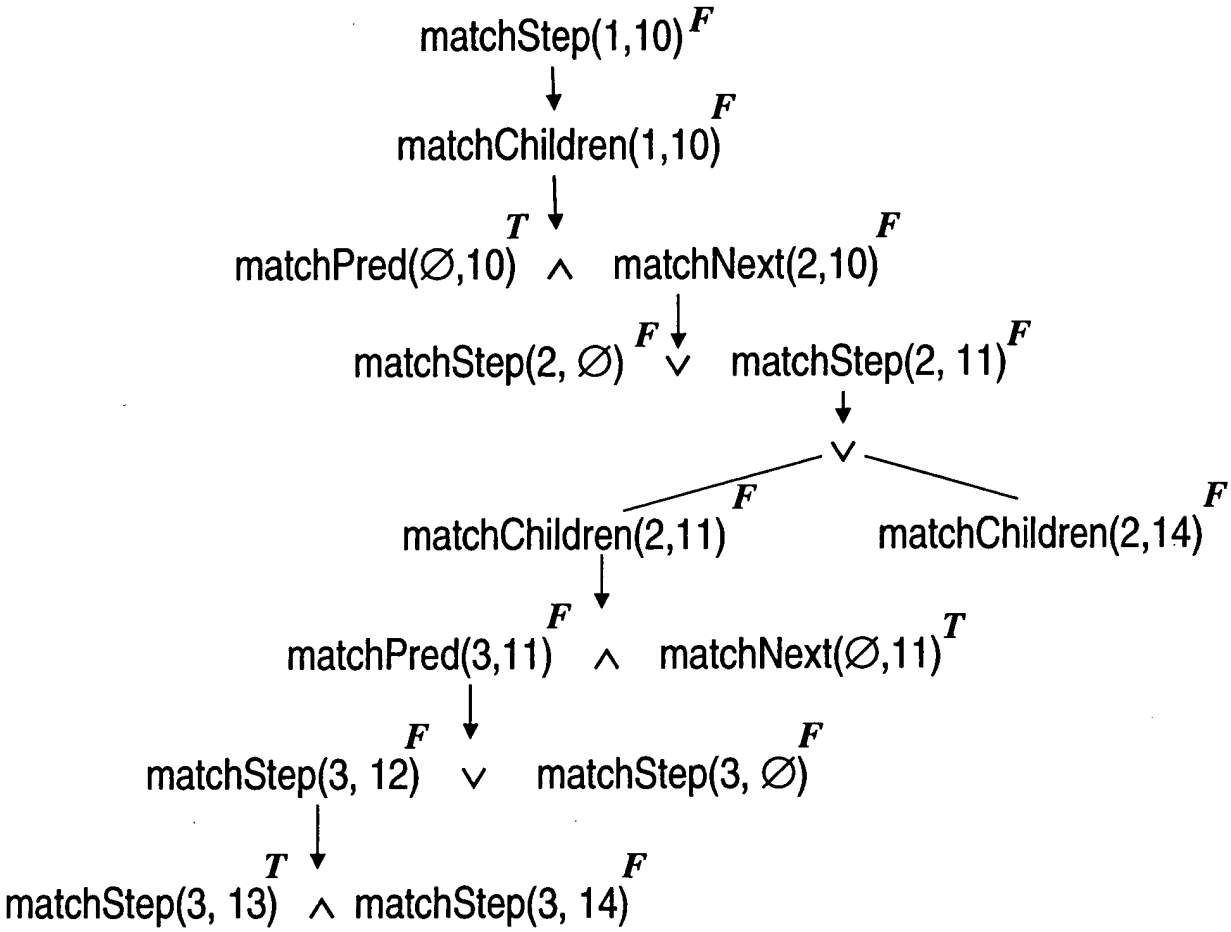

matchStep(1,10) F
matchChildren(1,10) F
matchPred(∅,10) T ∧ matchNext(2,10) F
matchStep(2, ∅) F ∨ matchStep(2, 11) F
∨
matchChildren(2,11) F
matchChildren(2,14) F
matchPred(3,11) F ∧ matchNext(∅,11) T
matchStep(3, 12) F ∨ matchStep(3, ∅) F
matchStep(3, 13) T ∧ matchStep(3, 14) F

FIG. 5

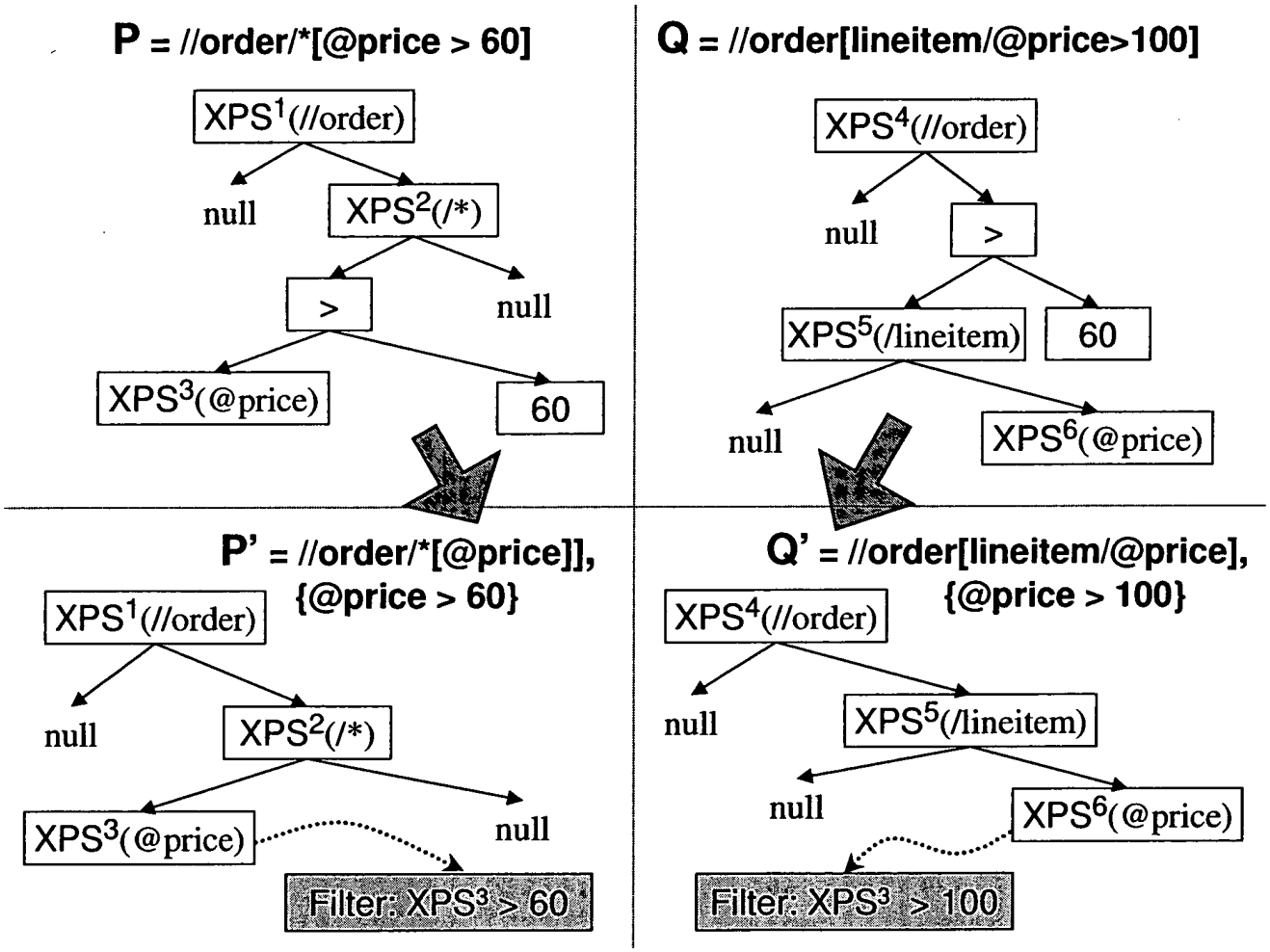
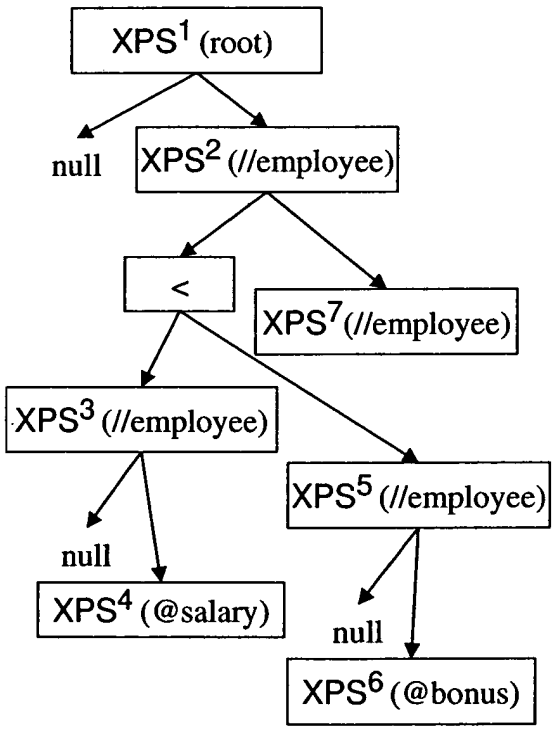

P = //order/*[@price > 60]
XPS1(//order)
null
XPS2(/*)
>
null
XPS3(@price)
60
Q = //order[lineitem/@price>100]
XPS4(//order)
null
>
XPS5(/lineitem)
60
null
XPS6(@price)
P' = //order/*[@price]],
{@price > 60}
XPS1(//order)
null
XPS2(/*)
XPS3(@price)
null
Filter: XPS3 > 60
Q' = //order[lineitem/@price],
{@price > 100}
XPS4(//order)
null
XPS5(/lineitem)
null
XPS6(@price)
Filter: XPS3 > 100

FIG. 6

```
ExtractPredicates(E)
foreach c in breadth-first traversal of E do
    if (c ∈ <, <, >, >, =, ≠) then
    // left extraction point
    lep = c.left;
    if (lep is an XPS) // traverse next's
        while (lep.next ≠ null) lep=lep.next;
    // right extraction point
    rep=c.right;
    if (rep is an PS) // traverse next's
        while (rep.next ≠ null) rep=rep.next;
    add "c (lep, rep)" to the filter list
    if (lep is a const) ∧ (rep is an XPS)
        replace c with the right child
    if (rep is a const) ∧ (lep is an XPS)
        replace c with left child
    if (lep is an XPS) ∧ (rep is an XPS)
        replace comparison in c with AND
```

FIG. 7

P = //employee[.//employee/@salary < .//employee/@bonus]//employee



Q = //employee[employee/@salary < employee[employee/@salary < employee/@bonus]/@bonus]//employee

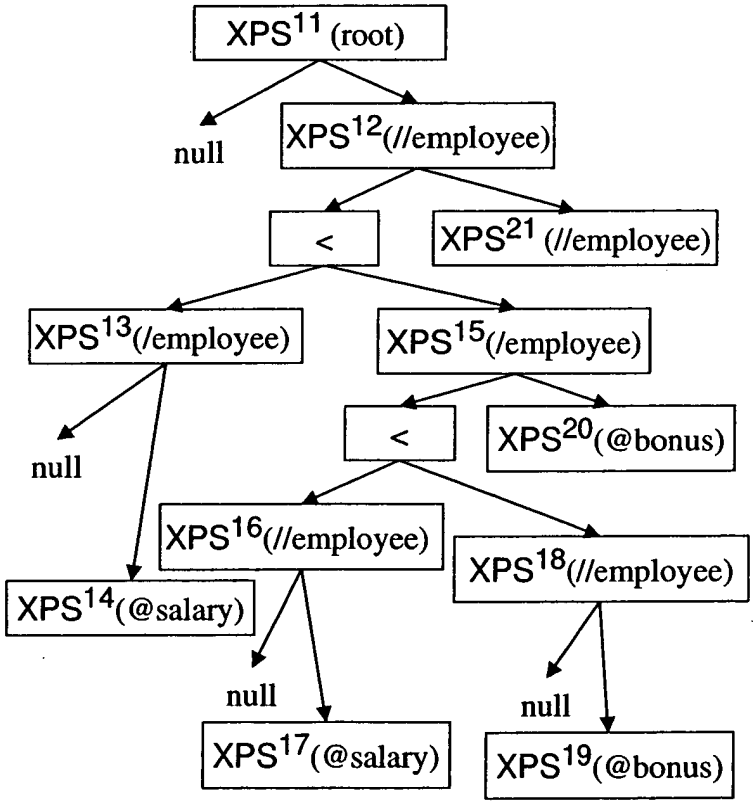


FIG. 8

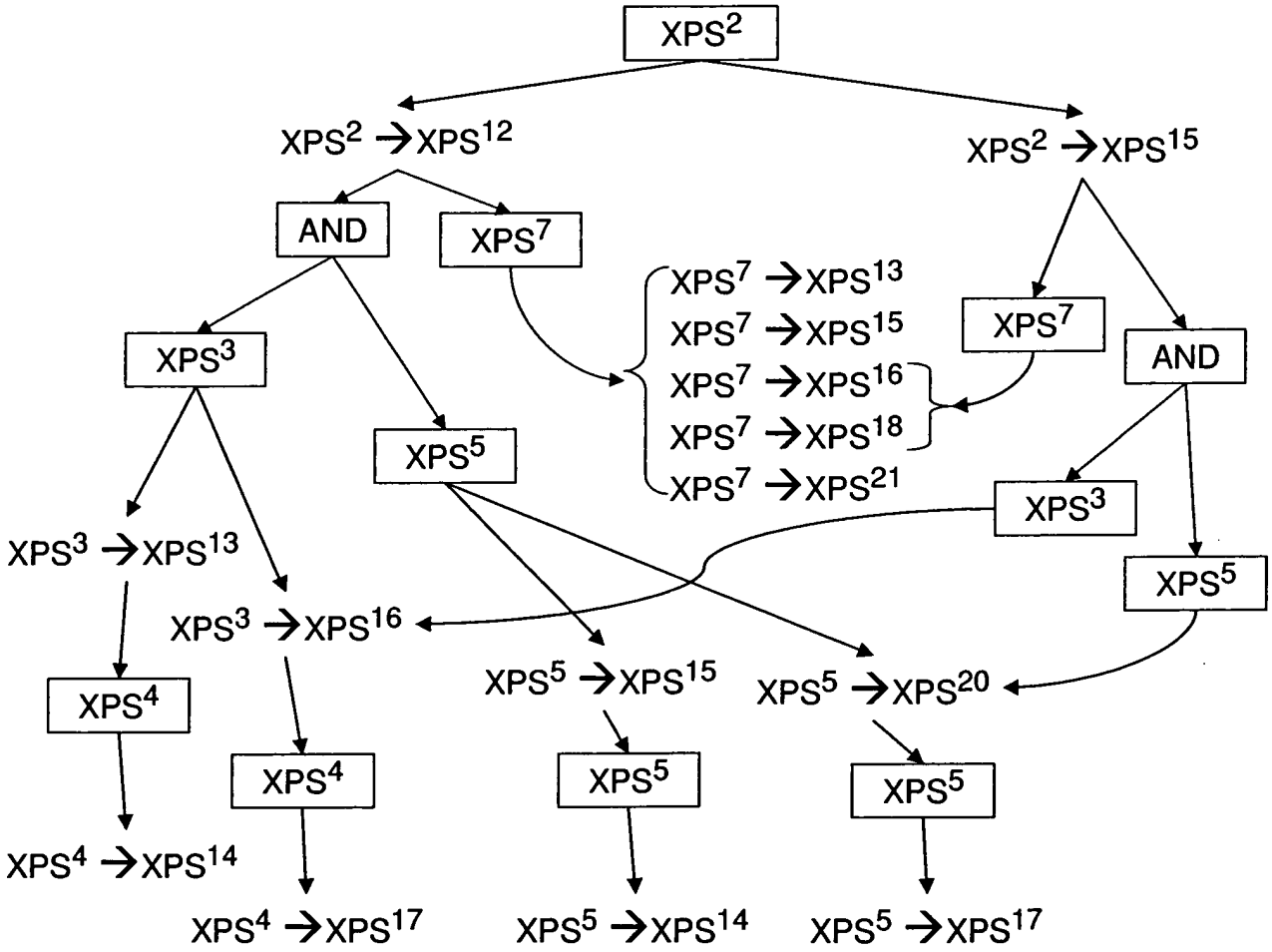

XPS2
XPS2 →XPS12
XPS2 →XPS15
AND
XPS7
XPS7 →XPS13
XPS7 →XPS15
XPS7 →XPS16
XPS7 →XPS18
XPS7 →XPS21
XPS7
AND
XPS3
XPS5
XPS3
XPS3 →XPS13
XPS5
XPS3 →XPS16
XPS5 →XPS15
XPS5 →XPS20
XPS4
XPS4
XPS5
XPS5
XPS4 →XPS14
XPS4 →XPS17
XPS5 →XPS14
XPS5 →XPS17

FIG. 9

REF/VALUE	PATH	TYPED VALUE
REF1	/A/B/C	5
REF2	/A/C	6
...	...	...

FIG. 10

Entry	*PXR content*	*Extra Navigation*	*Predicates*	`Name/Ancestors`
1	(NR)	Forward/Reverse	As residual	As residual
2	(DR)	Full Query	Full Query	Full Query
3	(copy)	Forward	Below QEP, as residual	No
4	(NR, path)	Forward/Reverse	As residual	As pushdown
5	(DR, path)	Full Query	Full Query	As pushdown
6	(copy, path)	Forward	Below QEP, as residual	As pushdown
7	(NR, value)	Forward/Reverse	As pushdown	As residual
8	(DR, value)	Full Query	As pushdown	Full Query
9	(copy, value)	Forward	As pushdown	Below QEP, as residual
10	(NR, path, value)	Forward/Reverse	As pushdown	As pushdown
11	(DR, path, value)	Full Query	Full Query	As pushdown
12	(copy, path, value)	Forward	As pushdown	As pushdown

FIG. 11

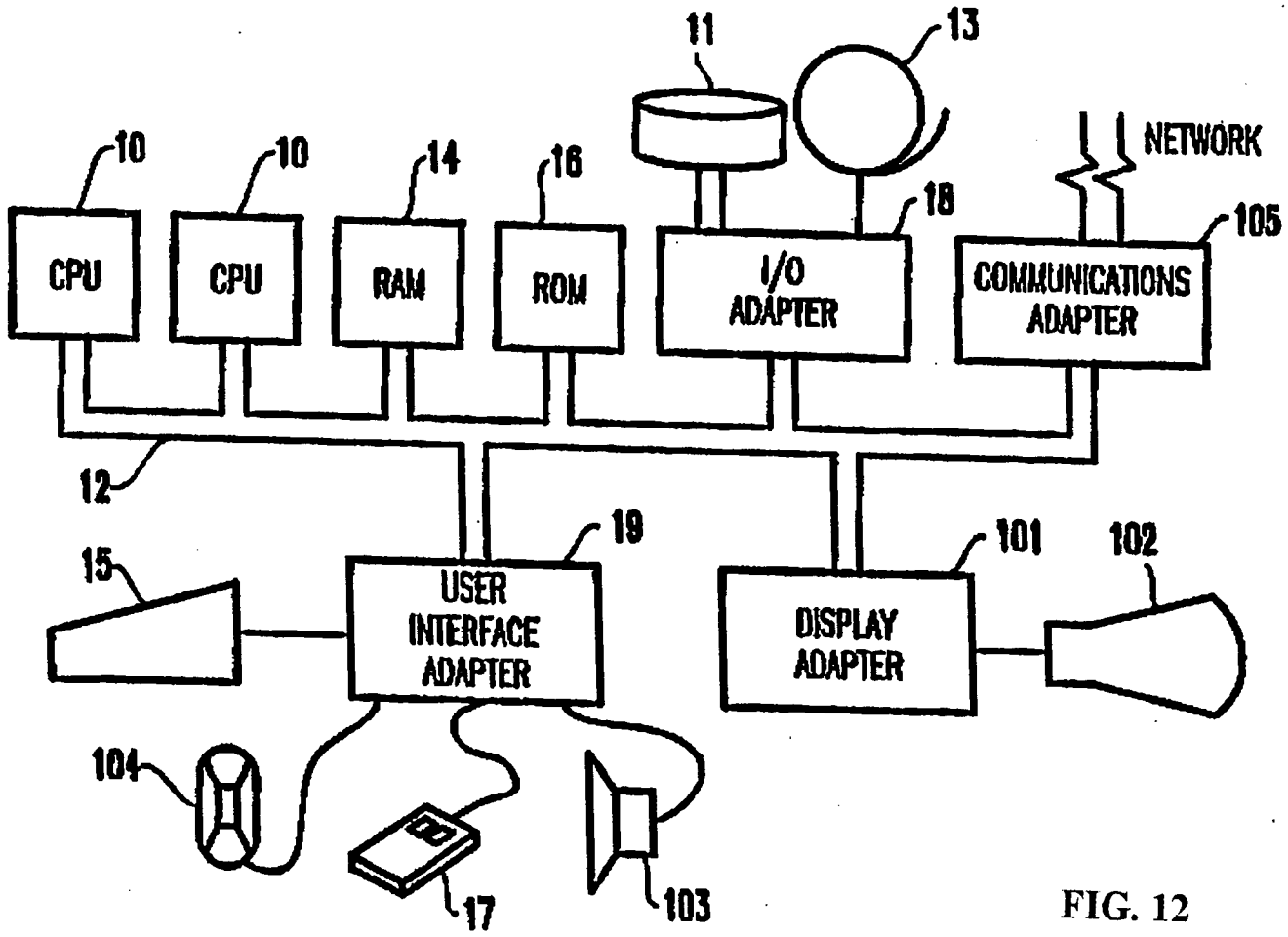

11
13
NETWORK
10
10
14
16
18
105
CPU
CPU
RAM
ROM
I/O ADAPTER
COMMUNICATIONS ADAPTER
12
19
101
102
15
USER INTERFACE ADAPTER
DISPLAY ADAPTER
104
17
103

FIG. 12